Hindawi
Security and Communication Networks
Volume 2021, Article ID 7206179, 16 pages
https://doi.org/10.1155/2021/7206179

Research Article

Publishing Triangle Counting Histogram in Social Networks Based on Differential Privacy

Tianzi Lv,[1,2] **Huanzhou Li**,[1,2] **Zhangguo Tang**,[1,2] **Fangzhou Fu**,[1,2] **Jian Cao**,[1,2] **and Jian Zhang**[1,2]

[1]*School of Physics and Electronic Engineering, Sichuan Normal University, Chengdu, Sichuan 610066, China*
[2]*Institute of Computer Network and Communication Technology, Sichuan Normal University, Chengdu, Sichuan 610066, China*

Correspondence should be addressed to Jian Zhang; zhangjian@sicnu.edu.cn

Received 19 August 2021; Revised 14 October 2021; Accepted 9 November 2021; Published 15 December 2021

Academic Editor: Hao Peng

The continuous expansion of the number and scale of social networking sites has led to an explosive growth of social network data. Mining and analyzing social network data can bring huge economic value and social benefits, but it will result in privacy leakage and other issues. The research focus of social network data publishing is to publish available data while ensuring privacy. Aiming at the problem of low data availability of social network node triangle counting publishing under differential privacy, this paper proposes a privacy protection method of edge triangle counting. First, an edge-removal projection algorithm TSER based on edge triangle count sorting is proposed to obtain the upper bound of sensitivity. Then, two edge triangle count histogram publishing methods satisfying edge difference privacy are given based on the TSER algorithm. Finally, experimental results show that compared with the existing algorithms, the TSER algorithm can retain more triangles in the original graph, reduce the error between the published data and the original data, and improve the published data availability.

1. Introduction

In recent years, social networking sites such as Weibo, WeChat, Facebook, LinkedIn, and Twitter have changed the way people communicate online. People use social networking sites to make friends, seek jobs, share resources, share life [1], and so on. The massive data collected in social networks has become a rich source of insight into fundamental societal phenomena, such as epidemiology, information dissemination, marketing [2], and so on. Most of the social network data is in the form of graphs. The publishing and analysis of these graphical data have enormous potential social benefits, such as providing more humanized services, publishing official statistics, and providing data sets for machine learning [3]. However, the attacker will infer from the data that the user's sensitive information has caused privacy leakage if the social network data is directly published. With the maturity of data mining technology, the problem of privacy leakage has become increasingly serious, and privacy protection has become a hot topic of general concern [4]. A large number of studies have provided privacy protection for social network data publishing. Sweeney et al. [5] first proposed the K-anonymity protection model, and then a series of data privacy protection technologies appeared, such as l-diversity [6], t-closeness [7], M-invariance [8], and so on. These models can prevent identity disclosure (such as ID, name, and other identifiers) but cannot resist background knowledge attacks [9]. Differential privacy model can successfully resist background knowledge attacks and has strict mathematical definitions, which is widely used in data publishing [10, 11].

The distribution query of subgraphs in social network data publishing is a research hotspot, including node and edge counting, triangle counting, k-triangle counting, k-stars counting, degree distribution, and node strength distribution [2]. As an important statistical characteristic of graphs, triangle counting plays a vital role in complex network analysis [12, 13]. It has been widely used in role identification and spam detection and is the protection focus of network graph data publishing.

It is a big challenge to reduce the sensitivity of query function and improve the availability of published data in differential privacy data publishing. Mapping the original graph to a new graph with lower sensitivity is the key to realizing and better utilizing differential privacy. The truncation method is usually used to reduce the sensitivity of differential privacy. The truncation method usually removes nodes or edges, projects the original graph to a new graph below the threshold, and preserves as much information of the original graph as possible during the projection [2]. Kasiviswanathan et al. [14] used differential privacy technology to study graph data for the first time. They show how to estimate the number of triangles satisfying edge differential privacy in social networks and how to calibrate the noise of subgraph counting. Do et al. [15] optimized the encryption matrix by modifying the existing security matrix calculation protocol to achieve multiparty secure triangular data transmission. Shoaran et al. [16] defined the group-based triangle metric in social networks and proposed a zero-knowledge privacy (ZKP) mechanism to provide privacy protection for data publishing. Ding et al. [17, 18] first proposed the privacy-preserving triangle counting problem of nodes and used the differential privacy of nodes to protect the triangle counting of large-scale graphs. They project the graph to a new graph with lower sensitivity to obtain the upper bounds of sensitivity in different distributions. Although the projection method reduces the query sensitivity, it leads to a large loss of original data and relatively low data availability. In addition, in order to achieve the set threshold of node triangle counting, a large number of edges need to be removed, and their proposed method greatly damages the original graph structure. Based on Ding et al., this paper proposes a privacy protection method for edge triangle counting.

The main goal of the triangle counting distribution publication is to publish approximate distribution as close as possible to the true triangle distribution of the original graph under the condition of differential privacy. For the publishing method of node triangle counting distribution, a large number of edges need to be removed to meet the node threshold, which leads to serious loss of graph data information and low data availability. Therefore, this paper proposes an edge triangle counting distribution publishing method, which can retain more graph information and improve data availability when meeting the edge threshold.

In order to further reduce the sensitivity of triangle counting publishing under differential privacy constraints, this paper proposes a new projection method. Based on this projection method, two histogram publishing mechanisms satisfying edge differential privacy are given. The main contributions of this paper can be summarized as follows:

(1) Firstly, the privacy-preserving triangle counting problem of edges is proposed, and a method to improve data availability while satisfying privacy protection is proposed. A triangle-count sort edge-removal algorithm (TSER) is proposed. Sort the edges according to the triangle count of the edges. Start processing from the edge with the largest triangle count, and remove the edge with the smallest triangle count in the set of edges that form a triangle with this edge. Finally, the edge triangle count is limited to a given threshold. This method can retain more triangles in the original graph while reducing the sensitivity, and the availability of published data is greatly improved.

(2) Based on the TSER projection method, two histogram publishing methods of edge-triangle count distribution under differential privacy are given: edge-triangle count distribution histogram TSER_His(G_θ) and edge-triangle count cumulative distribution histogram TSER_cumHis(G_θ), and it is proved that these two histogram publishing mechanisms meet the definition of edge differential privacy.

(3) Experimental results on different real data sets show that compared with the node triangle counting distribution publishing, the edge triangle counting distribution publishing method based on the TSER algorithm proposed in this paper can better retain the structural characteristics of the original graph and improve the availability of the published data.

The rest of this paper is organized as follows: Section 2 elaborates related work. Section 3 introduces some preliminary knowledge of social networks and differential privacy algorithms. Section 4 introduces the algorithm TSER proposed in this paper and gives two histogram publishing methods based on the TSER algorithm. Section 5 conducts experimental verification analysis and data utility measurement on the algorithm proposed in this paper. Section 6 summarizes this paper and looks forward to the next work.

2. Related Work

Differential privacy is a privacy protection model proposed by Dwork et al. [19]. By adding random noise to interfere with the sensitive information in the data, the purpose of privacy protection is achieved while maintaining the overall useful information of the data [20]. Based on the principle of data distortion, the differential privacy model provides strict mathematical theory support. Differential privacy can also successfully resist background attacks and is widely used in social networks, location privacy, recommendation systems, and other fields [2]. The main research issues of differential privacy include the nature of input and output data, mechanism design, and parameter setting. Data publishing based on differential privacy can be roughly divided into two categories: node differential privacy [14] and edge differential privacy [21].

In node differential privacy, only one node is different between two neighbor graphs. Any addition or deletion of one node has little effect on the query results. Liu et al. [22] studied node strength distribution based on node differential privacy, reducing sensitivity by limiting weight and degree. Qian et al. [23] proposed a privacy node strength publishing method based on edge differential privacy. Compared with the edge differential privacy of Qian et al., the node

differential privacy method of Liu et al. can provide stronger privacy protection, but it has low data availability. Zhang et al. [24] aimed at the problem of greater sensitivity of the node degree distribution under the definition of node differential privacy and proposed an edge-removal projection method SER based on degree ordering. Two histogram mechanisms of degree distribution under node difference privacy are given: SER-histogram and SER-cumulative histogram. Wu et al. [25] proposed two kinds of uncertain graph privacy protection algorithms for the privacy protection of social network graphs. The uncertain graph is a privacy protection method that converts the deterministic graph into the probability graph. The uncertain edge probability assignment algorithm based on differential privacy has high privacy protection and is suitable for the scenarios with high privacy protection requirements in social networks, but its data availability needs to be improved. Ding et al. [17, 18] first proposed the privacy-preserving triangle counting problem of nodes and proposed new projection algorithms DL and BA to obtain the upper bounds of sensitivity in different distributions. Two triangle counting distribution histograms of node differential privacy are given: triangle counting distribution histogram and cumulative distribution histogram. This projection algorithm reduces query sensitivity, but data processing causes a large loss of original graph information, and data availability is relatively low.

In the edge difference privacy, there is only one edge difference between the two neighbor graphs. Adding or deleting an edge between any two nodes in the graph has a negligible impact on the query results. Kasiviswanathan et al. [14] used differential privacy technology to study graph data for the first time, proposed to use the truncation method to project the original graph to reduce sensitivity, and showed how to estimate the number of triangles that meet edge differential privacy in social networks. Although this method can reduce the sensitivity, it deletes many unnecessary edges and loses a lot of effective information in the original graph. Sun et al. [26] proposed the k-triangle counting publication of composite graphs for the first time and proposed a composite graph publication method with edge difference privacy, which enables the published composite graphs to accurately respond to triangle counting queries with any constant k. For the degree distribution of social network graphs, Qian et al. [23] proposed that the degree distribution cannot fully represent the characteristics of nodes and graphs. They further studied the node strength histogram publication based on edge difference privacy, constructed a t-bound graph to limit the size of edge weight, and proposed two bucket clustering methods based on sequence and density to optimize the publication accuracy.

For the protection of triangle counting, there have been studies on the triangle counting protection for nodes, while the triangle counting protection for edges is still blank. Moreover, the projection algorithm of node triangle counting leads to large information loss of the original graph and low data availability. It is a big challenge to improve the availability of published data while ensuring differential privacy protection. In this paper, a new projection method is proposed to improve the data availability for the privacy protection triangle counting problem of edges.

3. Preparatory Knowledge

3.1. Social Network. Social networking is a virtual application platform created and managed by users on the Internet, providing information exchange, sharing, and news dissemination. The rich various applications of social networks provide people with great convenience, such as social communication, academic resource sharing, multimedia sharing, electronic payment, job search information, and so on. Social network data is usually mapped to graph structure (taking undirected unweighted graph as an example), as shown in Figure 1.

The social network data graph is represented by $G = (V, E)$, where V is the node and E is the edge. Nodes in network graph data refer to users or related institutions and organizations, and edges refer to the interaction between users and users, users and organizations, and organizations and organizations. The association between users is usually mutual; in general, the social network diagram is an undirected graph. Users are almost impossible to exist independently in social networks, more or less connected with the outside world.

Each element of nodes, edges, and network graph structures may involve private information in social networks. The privacy information of social networks can be divided into three parts: node privacy, edge privacy, and network structure privacy [9]. Social network node privacy includes node existence, node identity recognition, and node attribute information. Edge privacy mainly includes edge connection, edge weight, and edge attributes. Privacy of network graph structure usually refers to the analysis of the topological structure of social networks to obtain privacy information, such as degree distribution, path length, and subgraph structure. The subgraphs include triangles, k-triangles, and k-stars [2].

Publishing social network data may cause user privacy to be stolen by criminals and harm the interests of users. The main types of attacks on social networks are background knowledge attacks, reasoning attacks, active attacks, and passive attacks. Background knowledge attack means that the attacker has mastered some network structure or attribute information to identify the target of the attack. An attribute-oriented attack will identify the target through the privacy attributes of nodes or edges. Network structure-oriented attack mainly includes subgraph attack, node degree attack, and edge weight attack. In social networks, the most serious threat is background knowledge attacks because attacks often master a large amount of information in the network graph through some means to attack the target object.

Differential privacy can successfully resist background knowledge attacks and is widely used in the privacy protection of social networks.

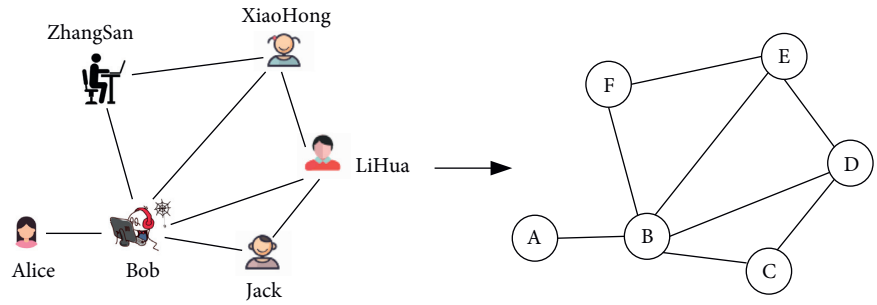


Figure 1: Social network structure graph.

3.2. Basic Knowledge of Differential Privacy

3.2.1. Differential Privacy

Definition 1 (neighboring data sets). Assuming that data sets D and D' have the same attribute structure, the symmetry difference between them is $D\Delta\ D'$; $|D\Delta\ D'|$ denotes the number of records in $D\Delta\ D'$, if $|D\Delta\ D'| = 1$, that is, there is only one record difference between D and D', which are called neighboring data sets, denoted by $D \cong D'$.

Various mapping functions of data set D are defined as query functions, and $f = \{f_1, f_2, \ldots\}$ is used to represent a set of query functions. Algorithm M processes the results of query f to achieve privacy protection conditions, which is called privacy protection mechanism for data set D.

Definition 2 (ε-differential privacy) [19]. For any two neighboring data sets D and D', there is a random algorithm M: $D \longrightarrow S^k$; algorithm M for all possible output results of D and D' is $S \in \mathrm{Rang}(M)$. If the algorithm M satisfies:

$$\Pr[M(D) \in S] \leq e^{\varepsilon}\Pr[M(D')S], \tag{1}$$

then algorithm M gives ε-differential privacy. Where probability $Pr[]$ is controlled by the randomness of algorithm M, also known as the risk of privacy being leaked. ε represents the privacy budget [27]. It can be seen from Definition 2 that the smaller ε is, the closer the output probability of algorithm M on the two neighboring data sets is, and the stronger the privacy protection is. However, the smaller e is, the greater the differential privacy noise will lead to the decrease of data availability.

3.2.2. Sensitivity. Differential privacy achieves the purpose of privacy protection by adding random noise to the query results to hide the difference between neighboring data sets. The sensitivity of query function refers to the maximum difference of query results on adjacent data sets, that is, the maximum change of query results when only one record in the data changes.

Definition 3 (sensitivity) [28]. Given a query function f, f: $D \longrightarrow R$, R is the set of all outputs of query function f on data set D. For any neighboring data sets D and D', the sensitivity of f is defined as follows:

$$\Delta f = \max_{D,D'} \left\| f(D) - f(D') \right\|_1. \tag{2}$$

The sensitivity Δf is only related to the query type, which determines the amount of disturbance noise. The higher the sensitivity, the greater the noise to be added and the lower the data availability.

3.2.3. Noise Mechanism. Noise mechanism is the main technology to achieve differential privacy protection. Laplace mechanism [28] and exponential mechanism [29] are commonly used to achieve differential privacy. The Laplace mechanism is suitable for the protection of numerical data, and the exponential mechanism is usually suitable for the protection of non-numerical data.

Definition 4 (exponential mechanism) [29]. Assuming $q(D, \varnothing)$ is the scoring function of data set D, which is used to measure the quality of output, and its sensitivity is Δq. If mechanism M satisfies ε-differential privacy, then it satisfies

$$M(D) = \left\{ q | \Pr[\phi \in \Phi] \infty \exp\left(\frac{\varepsilon q(D, q)}{2\Delta q} \right) \right\}. \tag{3}$$

Definition 5 (Laplace mechanism) [28]. Given a data set D, there is a function f: $D \longrightarrow R$, and its sensitivity is Δf. If the mechanism M satisfies ε-differential privacy, then it follows the following equation:

$$M(D) = f(D) + \mathrm{Lap}\left(\frac{\Delta f}{\varepsilon} \right), \tag{4}$$

where $\mathrm{Lap}(\Delta(f/\varepsilon))$ is random noise and obeys Laplace distribution with scale parameter $\Delta(f/\varepsilon)$. The mechanism shows that the size of the noise is related to the sensitivity of the query function f and the privacy budget ε.

Laplace mechanism is widely used in numerical data protection, so this paper uses the Laplace noise mechanism.

3.3. Node Differential Privacy and Edge Differential Privacy. Node differential privacy refers to arbitrarily adding or deleting a node in the graph, which has a very small impact on the query results. The definition of the neighboring graph in node differential privacy is as follows: for network graphs G and G', if a node is added or deleted in network

graph G to get G', then they are called neighboring graphs. In other words, graphs G and G' have only one node different, as shown in Figure 2. Node differential privacy can protect node attribute security, which makes attackers unable to speculate the existence of nodes in the network and has a strong privacy protection ability. When a node is randomly deleted/added, the worst case is that the node is connected to all the remaining nodes in the figure, and the query sensitivity of node differential privacy is relatively large.

Edge differential privacy means that adding or deleting an edge between any two nodes in the graph has a negligible impact on the query results. The definition of the neighbor graph in edge differential privacy is as follows: for two network graphs G and G', if the two graphs differ by only k edges, then G and G' are neighboring graphs, where $k = 1$ is the most widely used [21], as shown in Figure 3. Edge differential privacy focuses on protecting edge attribute privacy such as friendship, cooperation, trade, and trust, and the query sensitivity is relatively small.

The query sensitivity caused by changing a node is proportional to the size of the graph. For large-scale network graphs, the sensitivity of node differential privacy is often higher than that of edge differential privacy, so the added noise disturbance is large, which cannot guarantee sufficient data utility. Node differential privacy can provide stronger privacy protection, but the privacy protection of edge differential privacy has met the actual needs in most applications, especially in large-scale social networks, so the application of edge differential privacy is more extensive. Therefore, this paper protects the edge triangle count in social networks based on edge differential privacy.

3.4. Utility Metrics. In order to comprehensively evaluate the performance of different triangle counting distribution algorithms, this paper uses the following three evaluation indicators.

(1) Triangle retention ratio $\text{count}(T')/\text{count}(T)$. The triangle retention ratio can measure the loss of the social network graph structure caused by the projection algorithm, where $\text{count}(T)$ is the total number of triangles in the original graph and $\text{count}(T')$ represents the total number of triangles after projection. Generally speaking, the larger the value of $\text{count}(T')/\text{count}(T)$, the more triangles the projection algorithm can retain, and the higher the retention of the original image information, the better the algorithm effect.

(2) L_1 error [14]. L_1 error is the error between the histogram distribution of the graph data after adding noise and the histogram of the original graph data. The smaller the L_1 error is, the more similar the histogram before and after the algorithm is, and the higher the data availability is. For two edge triangle counting distributions D and D' with length M, the L_1 error between them is as follows:

$$L_1 = \sum_{i=0}^{M-1} \left| D_i - D'_i \right|. \tag{5}$$

While for the edge triangle counting distribution whose length is smaller than M, it is filled with 0 to M for convenience.

(3) KS distance [30]. KS distance is used to compare whether there are significant differences between the two empirical distributions, reflecting the proximity between the two histograms. The smaller the KS distance, the closer the histogram after adding noise is to the original histogram, and the higher the availability of the data. The KS distance is based on the cumulative distribution function. For the two triangle count distributions D and D', the KS distance between them is as follows:

$$\text{KS}(D, D') = \max \left| \text{CDF}_D(i) - \text{CDF}_{D'}(i) \right|, \tag{6}$$

where $\text{CDF}_D(i)$ is the cumulative distribution function value when the triangle count on the histogram distribution D is i.

4. Differential Privacy Edge Triangle Count Histogram Publication

Paired friendships were common in the early stages of social networking. With the development of the network, users establish more and more friendship relationships, thus forming more triangle relationships. The number of triangles in the social network can measure whether a network is mature, which is widely used in information dissemination, marketing, e-commerce recommendation, and so on. However, publishing the triangle count result directly will cause user privacy leakage. Before publishing the triangle count data, corresponding privacy protection measures are required.

4.1. Privacy Protection of Edge Triangle Counting. A simple model is used to briefly explain the triangle counting problem of edge privacy protection in the process of social network data publishing.

Given an undirected and unweighted social network structure, the social network has 10 user nodes and 17 relationship edges, as shown in Figure 4. The publishing and analysis of these graphic data have great potential social benefits. The data holder publishes the distribution histogram of edge triangle count, as shown in Figure 5. The abscissa of the histogram is the count of edge triangles, and the ordinate is the number of edges connected to x triangles.

Suppose there is an attacker who has strong background knowledge, and he knows the relationship edge of the social network except for E_6. The attacker is not sure whether there is a secret transaction between the two users connected by E_6. Once the attacker obtained the histogram of the edge triangle count distribution from the Internet, he found that one edge is connected to four triangles, and another edge is

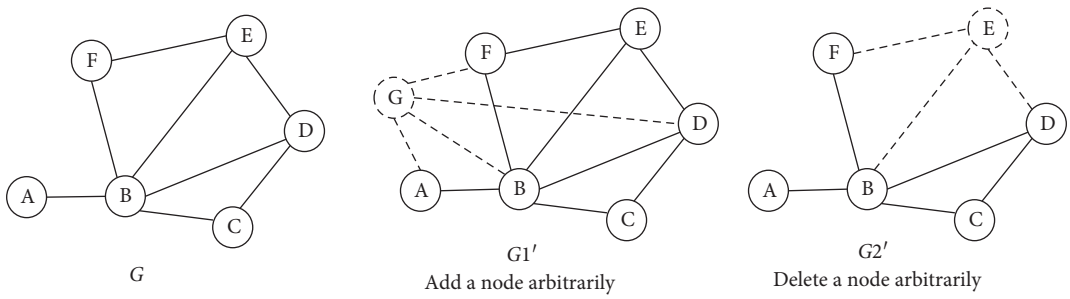


Figure 2: Node differential privacy neighboring graphs.

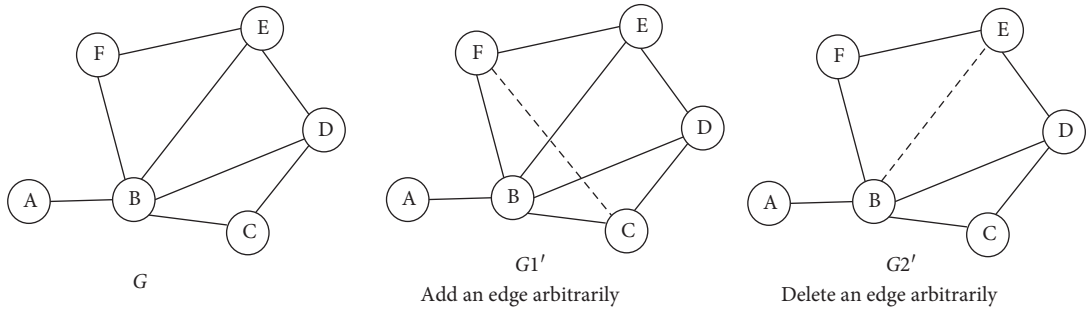


Figure 3: Edge differential privacy neighboring graphs.

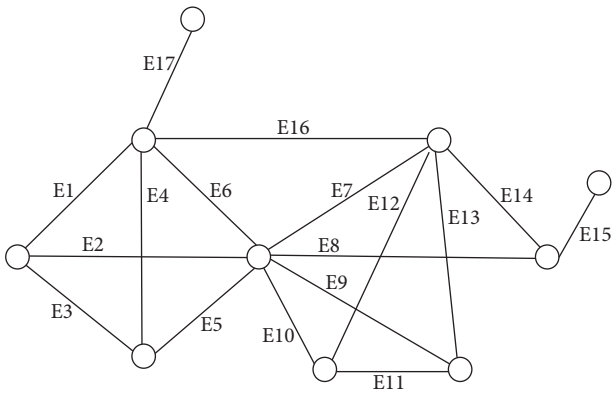


Figure 4: Simple social network structure.

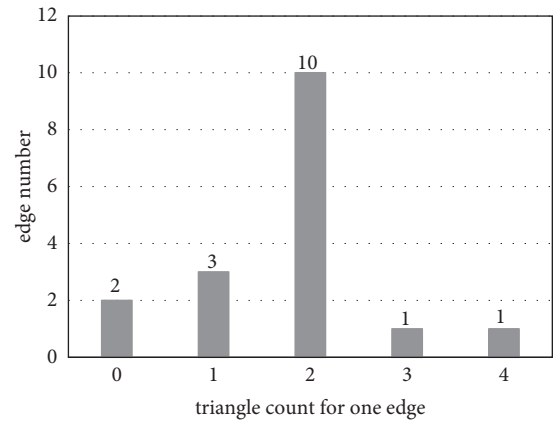


Figure 5: Edge triangle counting distribution histogram.

connected to three triangles. By comparing the background knowledge already mastered in the social network, the attacker can infer that E_6 must exist.

The number of edge triangles can also represent the number of mutual friends between the two users linked by that edge. It is easy for an attacker to mine the user's social privacy from the edge triangle count. Therefore, privacy protection measures are needed before publishing the edge triangle counting distribution. Differential privacy can resist background knowledge attacks and is widely used in data publishing. This paper studies the privacy protection of edge triangle counting based on edge differential privacy.

4.2. Proposed TSER Algorithm. This paper proposes a new graph projection algorithm in order to obtain the sensitivity upper bound of the triangle count distribution, which is called triangle-count sort edge-removal (TSER). This algorithm can retain more triangles in the projection process to improve the availability of published data. The algorithm is shown in Algorithm 1.

The algorithm first calculates the number of edge triangles of all edges in $G(V, E)$, records it in $E\mathrm{Tri}(i)$, and sorts the edges from large to small according to $E\mathrm{Tri}(i)$. The edges whose edge triangle count is greater than the threshold θ are recorded in the set $E(i)^+$, and then the edge $\mathrm{TriEdge}(E_i)$ forming the triangle with E_i is recorded. In $\mathrm{TriEdge}(E_i)$, each triangle marks an edge with the smallest number of edge triangles and is recorded in $\mathrm{minTriEdge}(E_i)$. For $\mathrm{minTriEdge}(E_i)$, the number of triangles is sorted from small to large, and then the edges with the smallest number of triangles are deleted successively until $E\mathrm{Tri}(i) = \theta$. Repeat

the above operation for each edge in $E(i)^+$ until all the edge triangles meet the threshold and end the algorithm.

In order to illustrate the process of Algorithm 1 more intuitively, Figure 6 shows examples of two graph projection methods. Setting the threshold $\theta = 2$, it can be seen from the original graph Figure 6(a) that the edge triangle counts of E_6 and E_7 are 3 and 4, respectively, both exceeding the threshold.

RER is a random edge-removal algorithm. Starting from the E_7 edge with the largest edge triangle count, the RER algorithm randomly removes E_9 and E_{12} to meet the threshold. For the E_6 edge, the RER algorithm randomly removes the E_2 edge to meet the threshold. The graph processed by the RER algorithm finally retains four triangles as shown in Figure 6(b).

The TSER algorithm used in this paper is shown in Figure 6(c). The edge triangle counts of E_6 and E_7 are greater than the threshold and are recorded as $E(i)^+$, and $E(i)^+$ is sorted from large to small according to the edge triangle count. Start preprocessing from the edge E_7 with the largest edge triangle count. Record that the edge forming the triangle with E_7 is TriEdge(E_7), and the edge with the smallest triangle count in each triangle is minTriEdge(E_7). Delete the edges E_{14} and E_{16} with the smallest edge triangle count in minTriEdge(E_7) in turn until E_7 meets the edge triangle count threshold 2. Update the whole graph and traverse the edges in $E(i)^+$ again. Repeat the above operations in turn and return the projected graph $G_\theta(V, E_\theta)$ when all edge triangle counts meet the threshold and end the algorithm. According to the TSER projection algorithm in this paper, eight triangles can be retained in the end.

Whatever the projection algorithm, the ultimate goal is to let the edge triangle count meet the given threshold θ and get the upper bound of sensitivity. From the example analysis of Figure 6, it can be seen that the RER algorithm may cause a large number of triangles to be lost when removing edges randomly. The TSER algorithm used in this paper can retain more triangles when meeting the threshold and approximate the original graph to the maximum extent, which reduces the error between the projection graph and the original graph so as to improve the data availability after differential privacy protection.

4.3. Differential Privacy Publishing Method Based on TSER. Based on the TSER projection algorithm, this paper proposes two histogram publishing methods: TSER-edge triangle counting distribution histogram TSER_His(G_θ) and TSER-edge triangle counting cumulative distribution histogram TSER_cumHis(G_θ).

4.3.1. TSER_His(G_θ). The edge triangle counting distribution histogram TSER_His$(G_\theta) = (\text{his}_1, \text{his}_2, \text{his}_3, \ldots)$ represents the number of edges whose edge triangles counting is i. It can be seen from Definition 3 that the sensitivity of query function refers to the maximum difference of query results on neighboring data sets, that is, when only one edge in G changes (deleting or adding one edge arbitrarily), it is the maximum difference between the histogram TSER_His(G) before the change and the histogram TSER_His(G') after the change. Given query function f, how many edges connect i triangles in graph G? According to Theorem 1, the sensitivity upper bound for publishing TSER_His(G_θ) is $4\theta + 1$.

Theorem 1. *For any neighboring data sets G and G' that differ by only one edge, there is:*

$$\text{TSER\_His}_i(G_\theta) - \text{TSER\_His}_i(G'_\theta)_I \le 4\theta + 1. \quad (7)$$

Proof. Assume, without loss of generality, graphs $G(V, E)$ and $G(V, E')'$ differ by only one edge E'. In the worst case, E' connects θ triangles in graph G'. When E' edge is deleted, θ triangles are removed. Removing 1 triangle affects at most 2 different edges except E', so removing θ triangles affect at most 2θ edges. And the change brought by each edge mapped to the histogram is 2, and the change brought by 2θ edges is 4θ. In addition, E' itself causes 1 change, so the total change brought about by deleting the E' edge to the histogram is the maximum $4\theta + 1$. Therefore, the sensitivity upper bound His(G_θ) of publishing TSER_His(G_θ) is $4\theta + 1$.

TSER-edge triangle counting distribution histogram TSER_His(G_θ) is shown in Algorithm 2.

4.3.2. TSER_cumHis(G_θ). The upper bound of the sensitivity of the TSER_His(G_θ) publishing method is $4\theta + 1$, which is still very high. In order to further reduce the sensitivity, another publishing method is proposed. The cumulative distribution histogram of edge triangle count TSER_cumHis$(G_\theta) = (\text{his}_1, \text{his}_2, \text{his}_3, \ldots)$ represents the number of edges whose edge triangle counting is less than or equal to i. Given query function f, how many triangles connected by edges in G are less than or equal to i? According to Theorem 2, the sensitivity upper bound for publishing TSER_cumHis(G_θ) is $2\theta + 1$.

Theorem 2. *For any neighboring data sets G and G' that differ by only one edge, there is*

$$\text{TSER\_cumHis}_i(G_\theta) - \text{TSER\_cumHis}_i(G'_\theta)_I \le 2\theta + 1. \quad (8)$$

Proof. Assume, without loss of generality, graphs $G(V, E)$ and $G'(V, E')$ differ by only one edge E'. In the worst case, E' connects θ triangles in graph G'. When E' edge is deleted, θ triangles are removed. When removing one triangle, it will affect at most two different edges except E', and removing θ triangles will affect at most 2θ edges. The change of each edge on the cumulative histogram is 1, and the change of 2θ edge is 2θ. In addition, E' itself causes 1 variation, so the maximum total variation of cumulative histogram is $2\theta + 1$ when E edge changes. It can be seen from the proof that the sensitivity upper bound cumHis(G_θ) of TSER_cumHis(G_θ) is $2\theta + 1$.

TSER-edge triangle counting cumulative distribution histogram publishing algorithm as shown in Algorithm 3.

Input: Original graph $G(V, E)$, threshold θ
Output: Truncated graph $G_\theta(V, E_\theta)$

```
(1)   for E_i ∈ E do
(2)   ETri(i) ⟵ Calculate the number of triangles connected to each edge
(3)   sort_ETri(i); / * E_i is sorted by ETri(i) from large to small [E_i, ETri(i)] * /
(4)   E(i)^+ ⟵ ETri(i) > θ; / * Edges with the number of triangles greater than the threshold * /
(5)   E(i) ⟵ ETri(i) ≤ θ; / * Edges whose number of triangles is less than threshold * /
(6)   while max ETri(i) > θ do
(7)   TriEdge(E_i) ⟵ Edges forming triangles with E(i)
(8)   TriEdge(E_i) ⟵ The smallest edge of ETri(i) in each triangle
(9)   sort_TriEdge(E_i)/ * E_i is sorted from small to large according to TriEdge(E_i) * /
(10)      for E_j ∈ TriEdge(E_i) do
(11)      if (ETri(j) == minTriEdge(E_i)^−)
(12)      Remove E_j, update ETri(i), E(i)^+, E(i)^−
(13)      Traverse E(i)^+ until all ETri(i) ≤ θ
(14)  Update the entire graph
(15)  Return G_θ(V, E_θ)
```

Algorithm 1: **TSER.** Edge-removal projection algorithm based on edge triangle counting sorting.

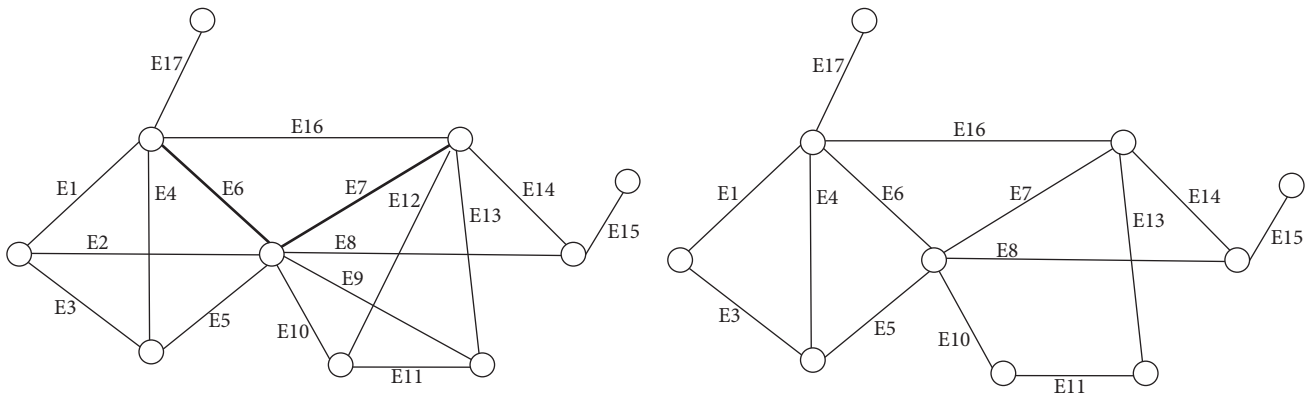


Figure 6: Continued.

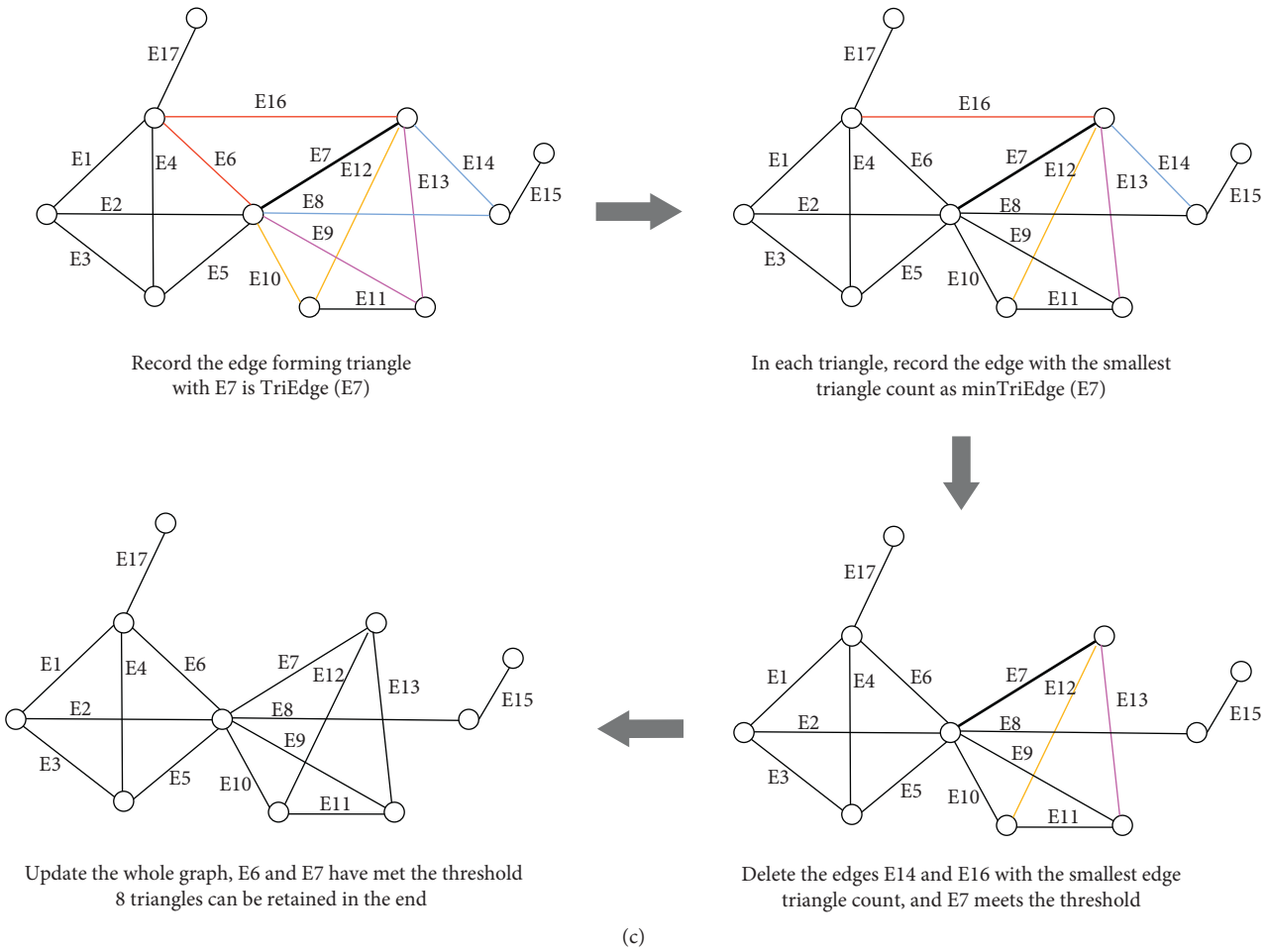


Figure 6: Examples of RER and TSER projection algorithm: (a) the original graph, (b) RER, and (c) TSER.

Input: Original graph $G(V, E)$, threshold θ, privacy budge ε
Output: Edge triangle counting distribution histogram under differential privacy TSER_His(G_θ)
(1) $G_\theta(V, E_\theta)$ preprocessed by Algorithm 1
(2) for $E_i \in E_\theta$ do
(3) $\text{his}_i(G_\theta) \longleftarrow$ Query: How many edges linked to i triangles?
(4) for $E_i \in E_\theta$ do
(5) TSER_His$(G_\theta) \longleftarrow \text{his}_i(G_\theta) + \text{Lap}(4\theta + 1/\varepsilon)$
(6) return TSER_His(G_θ)

Algorithm 2: TSER_His(G_θ). Edge triangle counting distribution histogram.

Input: Original graph $G(V, E)$, threshold θ, privacy budge ϵ
Output: Cumulative distribution histogram of edge triangle count under differential privacy TSER_cumHis(G_θ)
(1) $G_\theta(V, E_\theta)$ preprocessed by Algorithm 1
(2) for $E_i \in E_\theta$ do
(3) $\text{his}_i(G_\theta) \longleftarrow$ Query: How many edges linked to i triangles?
(4) for $\text{his}_i \in \text{his}(G_\theta)$ and $\text{his}_i < \theta$ do
(5) $\text{his}_i \neg \text{his}_i + \text{his}_{i-1}$
(6) for $E_i \in E_\theta$ do
(7) TSER_cumHis$_i(G_\theta) \longleftarrow \text{his}_i(G_\theta) + \text{Lap}(2\theta + 1/\varepsilon)$
(8) return TSER_cumHis(G_θ)

Algorithm 3: TSER_cumHis(G_θ). Cumulative distribution histogram of edge triangle counting.

5. Experiment and Analysis

This section uses real social network data sets to experiment with the proposed TSER projection algorithm and compares and analyzes the TSER_His(G_θ) and TSER_cumHis(G_θ) histogram publishing method with the previous algorithm.

5.1. Data Sets and Settings. Experiments use four real data sets from the Stanford Large Network Dataset Collection website [31]: (1) Facebook is an anonymous social data set; (2) Wiki-Vote is a Wikipedia voting network; (3) Cit-HepTh is a citation network for high-energy physics papers; and (4) Epinions is a who-trust-whom online social network of a general consumer review site Epinions.com. Members of the site can decide whether to "trust" each other. The amount of data in these four data sets is from small to large. Detailed statistics of each data set are shown in Table 1. $\mathrm{Tri}_{\mathrm{sum}}$ is the sum of triangles in the data set; $E\mathrm{Tri}_{\mathrm{max}}$ is the max number of edge triangle count; and $E\mathrm{Tri}_{\mathrm{avg}}$ is the average of all edge triangle counts. In this paper, all directed graphs are preprocessed into undirected. All experiments were run on a PC with AMD Ryzen 74800H with Radeon graphics CPU and a 64 bit Windows operating system.

In this paper, the threshold θ is selected according to the reference sequence $\{1, 2, 4, 8, \ldots, 2^{[2\log_2(|E|)]}\}$ [32, 33], and the value of privacy budget ε is between 0.5 and 1.5 [22, 23]. Due to the randomness of Laplacian noise and in order to better measure the algorithm performance, for each algorithm, we use the mean error of 100 runs. Experiment 1 compares the count(T')/count(T) index of the proposed edge triangle counting method and the node triangle counting method [17]. Experiment 2 compares the L_1 error and KS distance of TSER_His(G_θ), TSER_cumHis(G_θ), BA_cumHis(G_θ), and BA_cumHis(G_θ) running on different data sets.

5.2. Analysis of Experiment Results. Experiment 1. Triangle retention ratio count(T')/count(T) of edge triangle and node triangle publishing methods.

Triangle retention ratio count(T')/count(T) can measure the loss of the social network graph structure caused by the projection algorithm. The higher the count(T')/count(T) is, the more the original graph information is retained, and the better performance is. The experimental results are shown in Figure 7.

(The $E\mathrm{Tri}_{\mathrm{max}}$ of the Facebook data set is less than 512, so its threshold θ is set to 256 in the experiment.)

It can be seen from the experimental results in Figure 7:

(1) On different thresholds of all data sets, the triangle retention ratio count(T')/count(T) of edge triangle counting publishing method proposed in this paper is much better than the node triangle counting publishing method. This is because the number of triangles connected to nodes is much more than that of edges connected, so in order to reach the threshold, the node triangle count publishing method often needs to delete a large number of edges in the projection process, resulting in a large reduction in the number of triangles in the graph, low retention of the original graph structure information, and poor data availability. In contrast, when the edge triangle counting meets the threshold, fewer edges are removed, and more triangles are retained. Therefore, the edge triangle counting distribution method proposed in this paper can better retain the original graph information.

(2) For the same data set, the larger the threshold θ is, the higher the triangle retention ratio count(T')/count(T) is, and the integrity of the original graph is better guaranteed. However, with the increase of θ, it can be seen from Theorems 1 and 2 that the upper bound of sensitivity will also increase, and the differential privacy noise will also increase. Therefore, it cannot be said that the greater the θ value is, the better. In practical applications, it is necessary to set the θ value according to the characteristics of the data set itself and the different requirements of the availability and privacy of the published data.

Experiment 2. Evaluation of L_1 error and KS distance for TSER_His(G_θ), TSER_cumHis(G_θ), BA_cumHis(G_θ) and DL_cumHis(G_θ).

Reference [18] proposed an effective strategy to reduce the loss of graph data information, which is called best adaptation (BA). BA algorithm first removes the edge with the largest triangle count to meet the node threshold. BA_cumHis(G_θ) is a cumulative histogram publishing method based on the BA. DL projection algorithm [17] is to delete edges from the linked nodes that have a larger degree. DL_cumHis(G_θ) is a cumulative histogram publishing method based on the DL.

L_1 error and KS distance are used to comprehensively measure the histograms of TSER_His(G_θ), TSER_cumHis(G_θ), and BA_cumHis(G_θ). The smaller the L1 error and KS distance, the smaller the difference between the histogram of the differential privacy protection and the original histogram, and the higher the data availability. The threshold 128 is a commonly used intermediate value to better accommodate both large and small data sets. (The next experiment will discuss the impact of θ on data availability.) When the threshold θ is set to 128, the effect of different privacy budget ε on the publication histogram is shown in Figure 8.

Figure 8 shows the curves of L_1 error and KS distance of each publishing method changing with the value of ε on four data sets. The experimental results show that:

(1) For different data sets and different ε, the L_1 error and KS distance of the two publishing methods of TSER_His(G_θ) and TSER_cumHis(G_θ) in this paper are much smaller than those of the BA_cumHis(G_θ) and DL_cumHis(G_θ). It shows that the publishing method based on the TSER projection in this paper is closer to the original data, which obviously improves the data availability.

Table 1: Information of four data sets.

Data set	$\|V\|$	$\|E\|$	$\mathrm{Tri}_{\mathrm{sum}}$	$E\mathrm{Tri}_{\mathrm{max}}$	$E\mathrm{Tri}_{\mathrm{avg}}$
Facebook	4,039	88,234	1,612,010	293	54
Wiki-Vote	7,115	103,689	608,389	562	19
Cit-HepTh	27,770	352,807	1,478,735	1572	12
Epinions	75,879	508,837	1,624,481	603	15

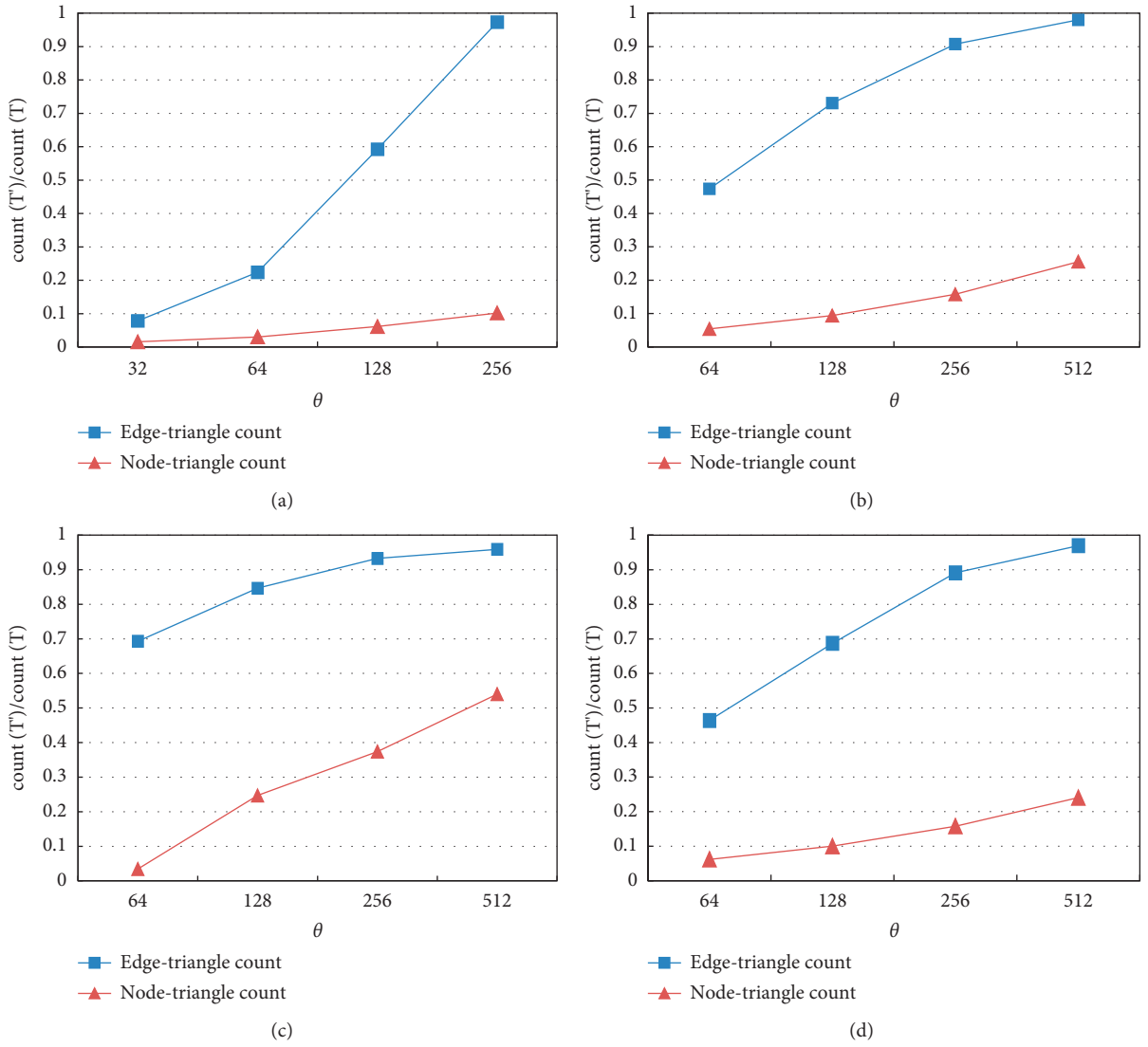
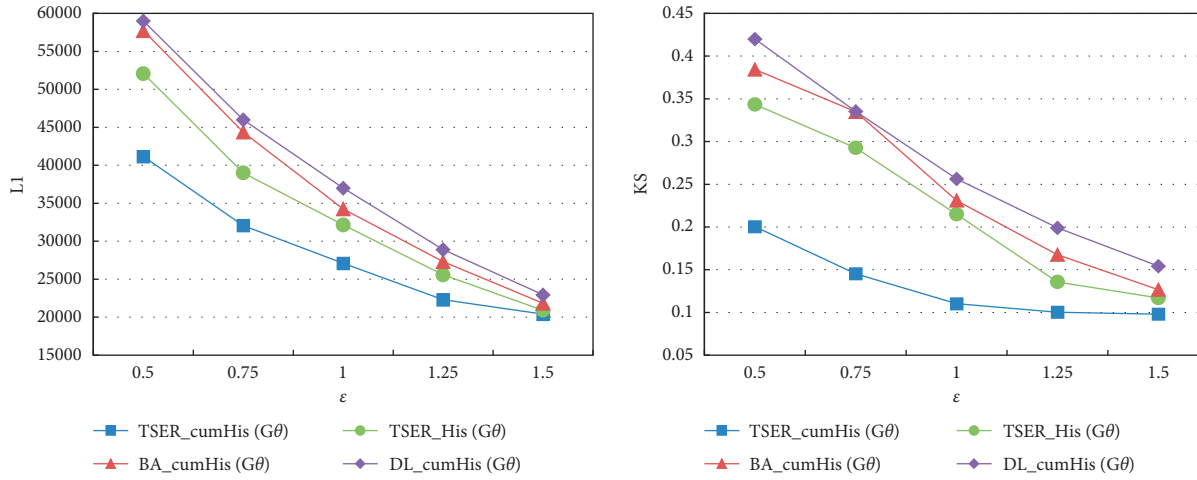


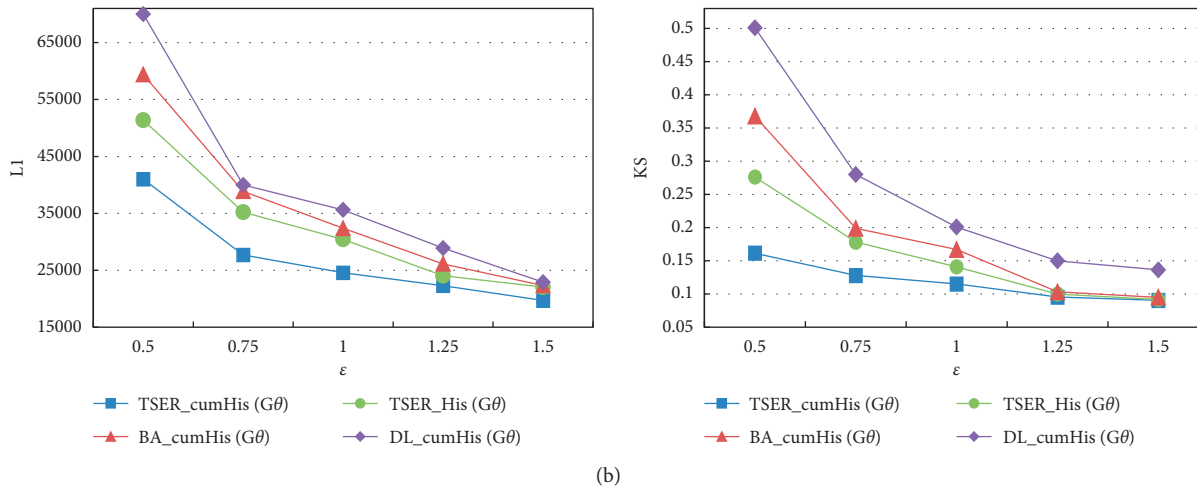
Figure 7: Comparison of count(T')/count(T) on different data sets: (a) Facebook, (b) Wiki-Vote, (c) Cit-HepTh, and (d) Epinions.

(2) Comparing the three cumulative histograms of TSER_cumHis(G_θ), BA_cumHis(G_θ), and DL_cumHis(G_θ), the L_1 error and KS distance of TSER_cumHis(G_θ) are both lower than BA_cumHis(G_θ) and DL_cumHis(G_θ). The reason may be that the BA and DL projection algorithm deletes too many edges when meeting the threshold and loses a lot of effective information in the original image, so the error of the histogram is too large. Among the three publishing algorithms, TSER_cumHis(G_θ) algorithm works best.

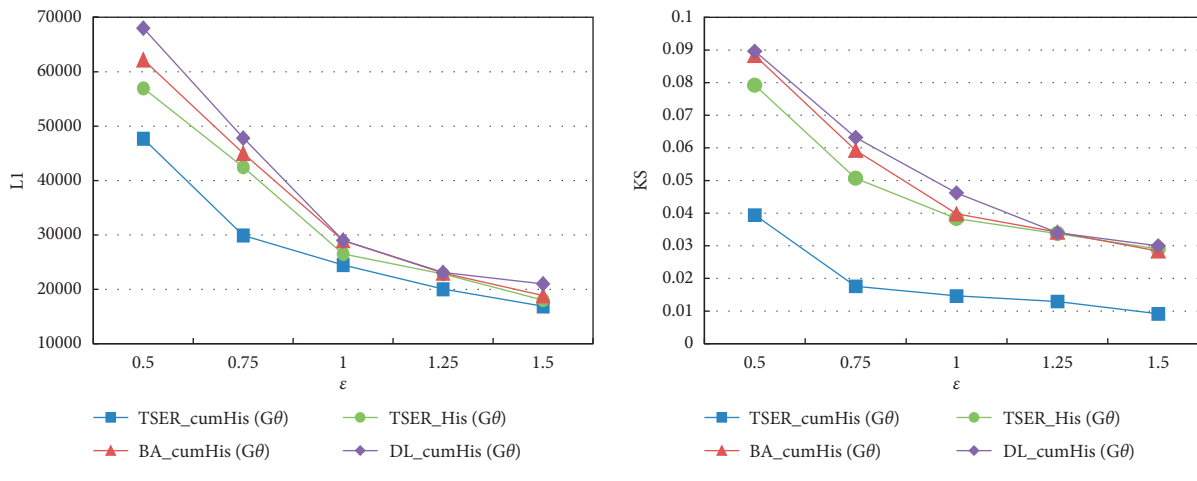
(3) Comparing the two publishing methods TSER_His(G_θ) and TSER_cumHis(G_θ) proposed in this paper, the experimental analysis of different data sets and different ε shows that the L_1 error and KS distance of TSER_cumHis(G_θ) are much smaller than TSER_His(G_θ). It can be explained by Theorems 1 and 2 that the sensitivity of the distribution histogram is higher than that of the cumulative histogram, resulting in a large amount of noise added, a large error in the histogram, and relatively low data availability. So TSER_cumHis(G_θ) has a



(a)

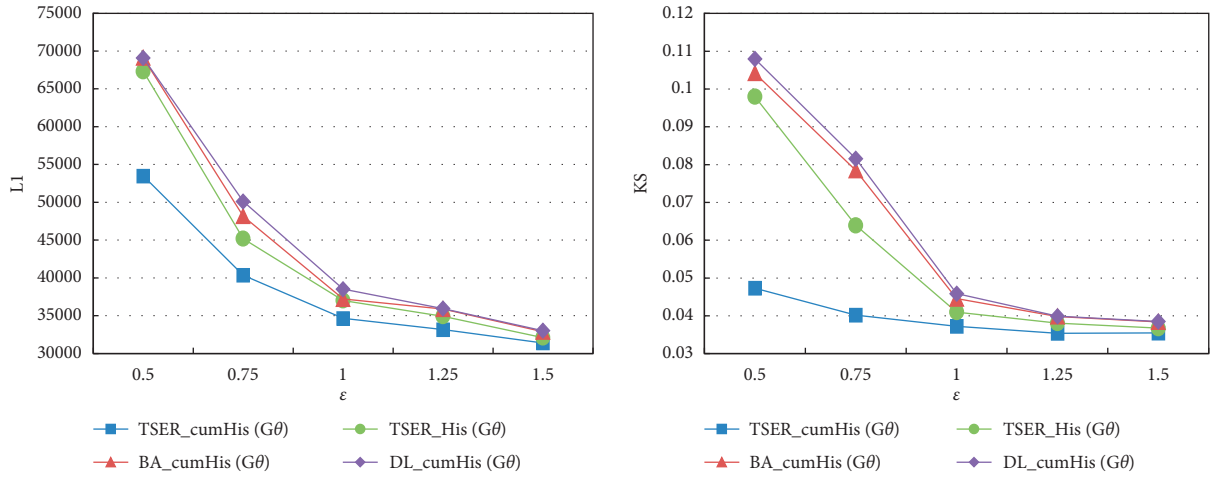


(b)



(c)

Figure 8: Continued.



(d)

FIGURE 8: Changes of L_1 error and KS distance with different ε values: (a) Facebook, (b) Wiki-Vote, (c) Cit-HepTh, and (d) Epinions.

smaller error and can better describe the original graph information.

(4) Comparing L_1 error and KS distance of the same data set, the smaller the privacy budget ε is, the larger the L_1 error and KS distance are, which is consistent with Definition 2 of differential privacy. The smaller the privacy budget ε, the stronger the privacy protection, but the greater the amount of corresponding noise, the greater the histogram error. In practical applications, the privacy budget ε should be set according to the security protection requirements and data availability needs of different data. If publishing data require high-strength privacy protection, the privacy budget ε is set to be smaller, but the data availability is relatively low. If the data requires better availability and lower privacy protection requirements, the privacy budget can be set a bit larger. In short, privacy budget ε should be reasonably set on the premise of ensuring data privacy security.

Through the above analysis, compared with the existing algorithms, the edge triangle count publishing method proposed in this paper can better retain the original graph information while ensuring data privacy and improve data availability. The effect of TSER_cumHis (G_θ) publishing method is better than that of TSER_His (G_θ), BA_cumHis (G_θ), and DL_cumHis (G_θ).

Threshold θ is also an important parameter; the value of θ will affect the availability of data. When ε is set to 1, the effect of different θ on the publishing method is shown in Figure 9.

(The $E\text{Tri}_{\max}$ of the Facebook data set is less than 512, so its threshold θ is set to 256 in the experiment.)

Figure 9 shows the curves of L_1 error and KS distance changing with the value of θ on each data set. It can be seen from the figure:

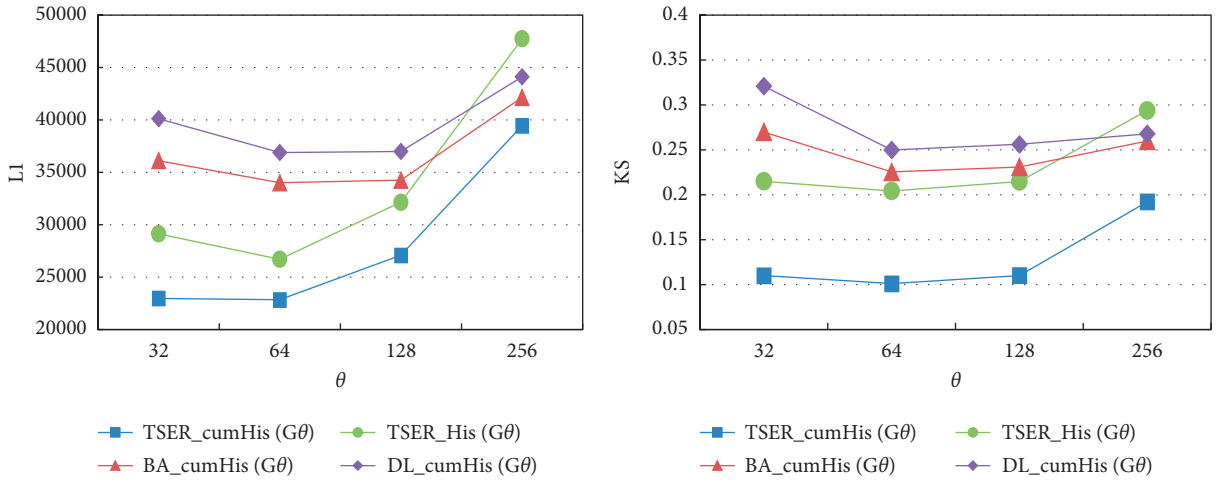
(1) In terms of different data sets and different θ values, the L_1 error and KS distance of TSER_cumHis (G_θ) algorithm in this paper both are the smallest, indicating that the publishing method in this paper is closer to the original data and improves the availability of publishing data.

(2) The value of θ has great differences in the impact of different types and scales of data sets. The L_1 error and KS distance curves of the two small data sets of Facebook and Wiki-Vote are generally similar as the trend of θ value changes, while the change curves of the two large data sets of Cit-HepTh and Epinions are similar.

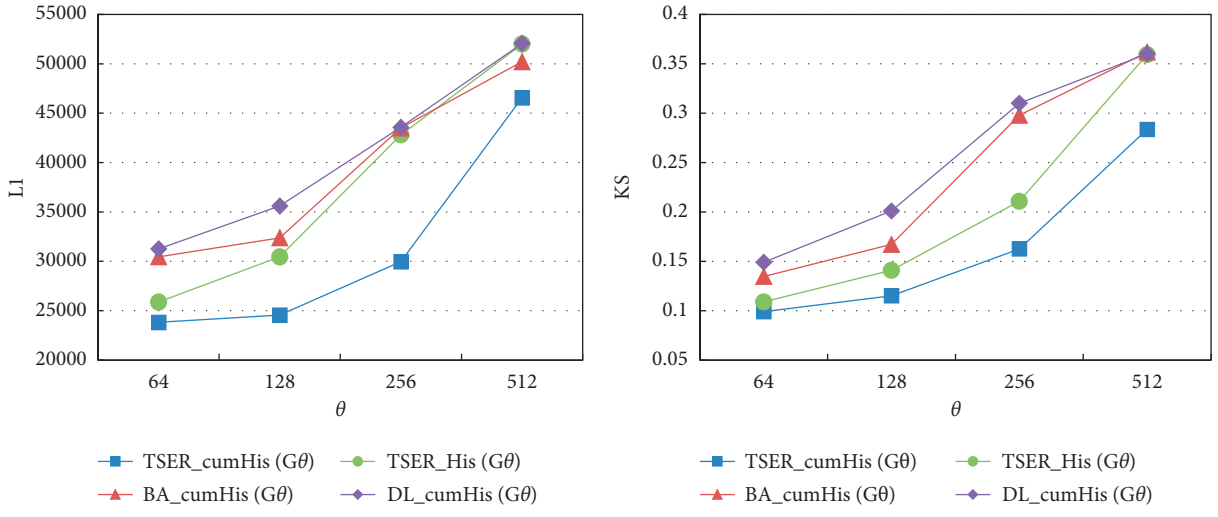
(3) For a relatively small data set such as Wiki-Vote, the curve of L_1 error and KS distance increases as θ value increases. For these small data sets, when the value of θ is larger, although more original image information can be retained, it also results in greater noise, increased L_1 error and KS distance, and low data availability. Therefore, the error of the histogram in this case is mainly caused by large noise.

(4) In large data sets such as Epinions, the values of L_1 error and KS distance are maximum when θ is 64. If the θ value is set to be small, the resulting noise is also small. However, the too small threshold θ will cause a large number of edges to be deleted during graph projection, which severely destroys the graph structure and loses a large amount of information, resulting in large errors in publishing histograms and low data availability.

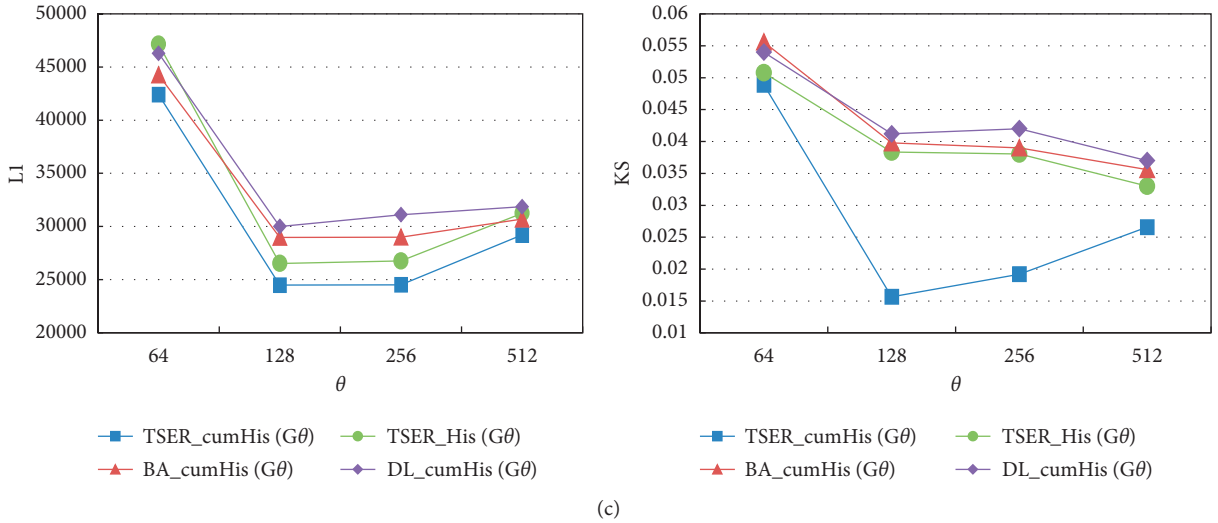
From the above analysis, it can be seen that the threshold θ has a significant impact on different publishing methods, different types, and scales of social network data sets. Large noise will cause larger errors in small data sets, so a smaller threshold θ can be set to improve data availability. In the process of large-scale social network data publishing, if the threshold is set to be small, a large amount of information will be lost in the process of graph projection. Therefore, a relatively large threshold can be set to improve data availability.



(a)



(b)



(c)

Figure 9: Continued.

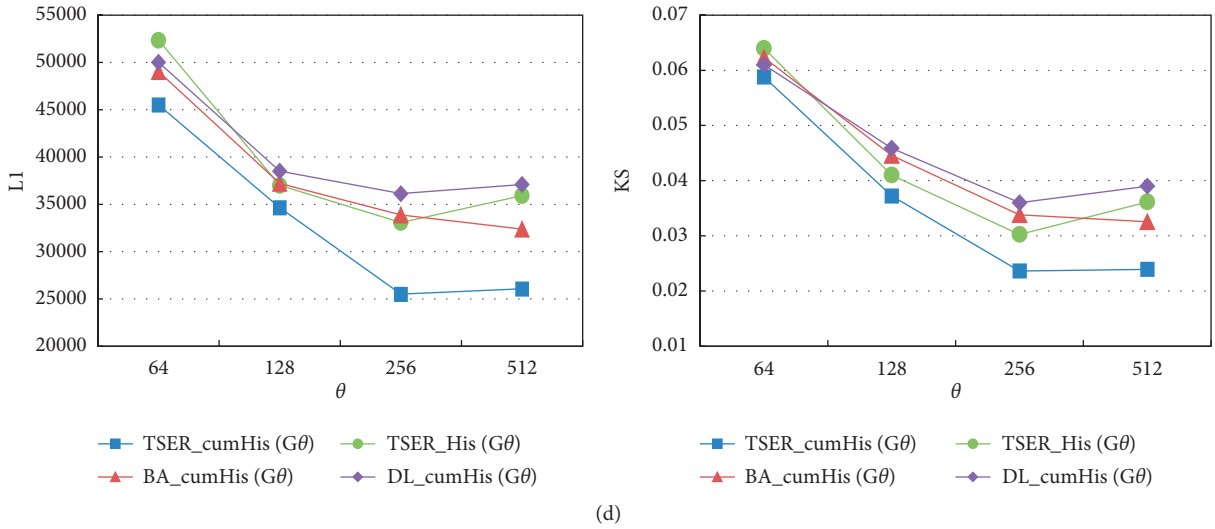


FIGURE 9: Changes of L_1 error and KS distance with different θ value: (a) Facebook,.(b) Wiki-Vote, (c) Cit-HepTh, and (d) Epinions.

Based on the analysis of the results of Experiments 1 and 2, the publishing method in this paper can reduce the error between the published data and the original data while protecting privacy and improve data availability. The values of the privacy budget ε and the threshold θ will affect the availability of published data. Therefore, in complex application scenarios, it is necessary to set the values of ε and θ reasonably according to different privacy requirements, different types and sizes of data sets, and different application scenarios. It is crucial to comprehensively weigh privacy protection and data availability and find the best balance between privacy protection and data availability.

6. Conclusions and Future Work

Privacy protection has been paid more and more attention in academia, industry, and daily life, and the privacy protection of social networks has become a hot topic. How to publish available data while protecting privacy has become one of the significant issues to be solved urgently. Differential privacy satisfies strict mathematical definitions and can resist the background knowledge attack that is one of the vital technologies of social network privacy protection.

This paper mainly studies the privacy protection of subgraphs in social networks. Firstly, the privacy protection of edge triangle counting is proposed, and an edge-removal projection algorithm (TSER) based on edge triangle sorting is proposed. Then, the original graph is projected by the TSER algorithm to obtain the upper bound of the sensitivity of the published data. Finally, based on the TSER projection algorithm, two edge triangle counting histogram publishing methods satisfying edge differential privacy are given. The experiments on real data sets show that compared with the existing algorithms, the TSER projection algorithm proposed in this paper can retain more triangle information in the original graph. The cumulative histogram publishing method based on TSER has advantages in both L_1 error and KS distance, which makes the published histogram closer to the original edge triangle counting distribution and improves the data availability. Since this paper only aims at unweighted social networks, the next step is to focus on the privacy protection of weighted social networks and extend the algorithm to weighted social networks.

Data Availability

The data can be obtained from https://snap.stanford.edu/data/#socnets.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (grant number: U1836103).

References

[1] A. M. V. Venkata Sai and Y. Li, "A survey on privacy issues in mobile social networks," *IEEE Access*, vol. 8, Article ID 130921, 2020.

[2] T. Q. Zhu, G. Li, W. L. Zhou, and P. S. Yu, "Differentially private data publishing and analysis: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1619–1638, 2017.

[3] J. H. Abawajy, M. I. H. Ninggal, and T. Herawan, "Privacy preserving social network data publication," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 3, pp. 1974–1997, 2016.

[4] Y. Fu, Y. Yu, and X. Wu, "Differential privacy protection technology and its application in big data environment," *Journal on Communications*, vol. 40, no. 10, pp. 157–168, 2019.

[5] L. Sweeney, "K-Anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, p. 557, 2002.

[6] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: privacy beyond

k-anonymity," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*, p. 24, Atlanta, GA, USA, April 2006.

[7] N. Li, T. Li, and S. Venkatasubramanian, "Closeness: privacy beyond k-Anonymity and l-Diversity," in *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering*, pp. 106–115, Istanbul, Turkey, April 2007.

[8] X. Xiao and Y. Tao, "M-invariance: towards privacy preserving re-publication of dynamic datasets," in *Proceedings of the International Conference on Management of Data*, Beijing, China, June 2007.

[9] X. Y. Liu, B. Wang, and X. C. Yang, "Survey on privacy preserving techniques for publishing social network data," *Journal of Software*, vol. 25, no. 3, pp. 576–590, 2014.

[10] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth, "The reusable holdout: preserving validity in adaptive data analysis," *Science*, vol. 349, no. 6248, pp. 636–638, 2015.

[11] X. Ping, Z. T. Qing, and W. X. Feng, "A survey on differential privacy and applications," *Chinese Journal of Cumputers*, vol. 37, no. 1, pp. 101–122, 2014.

[12] W. Zhuo, S. Bo, and P. Wei, "Parallel algorithm for triangle enumeration," *Journal of Computer Applications*, vol. 37, no. 12, pp. 3397–3400, 2017.

[13] J. Hongqiao and D. Yihong, "Research progress of triangle counting in big data," *Telecommunications Science*, vol. 32, no. 6, pp. 153–162, 2016.

[14] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith, "Analyzing graphs with node differential privacy," in *Proceedings of the Theory of Cryptography*, pp. 457–476, Tokyo, Japan, March 2013.

[15] H. G. Do and W. K. Ng, "Privacy-preserving triangle counting in distributed graphs," in *Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 917–924, Crans-Montana, Switzerland, May 2016.

[16] M. Shoaran and A. Thomo, "Editorial: zero-knowledge-private counting of group triangles in social networks," *The Computer Journal*, vol. 60, no. 1, pp. 126–134, 2017.

[17] X. Ding, X. Zhang, Z. Bao, and H. Jin, "Privacy-preserving triangle counting in large graphs," in *Proceedings of the The 27th ACM International Conference on Information and Knowledge Management*, pp. 1283–1292, Torino, Italy, October 2018.

[18] X. Ding, S. Sheng, H. Zhou et al., "Differentially private triangle counting in large graphs," *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2021.

[19] C. Dwork, "Differential privacy," *Automata, Languages and Programming*, vol. 4052, pp. 1–12, 2006.

[20] C. Dwork, "A firm foundation for private data analysis," *Communications of the ACM*, vol. 54, no. 1, pp. 86–95, 2011.

[21] K. Vishesh, Y. Grigory, S. Adam, and R. Sofya, "Private analysis of graph structure," *ACM Transactions on Database Systems*, vol. 4, no. 3, pp. 1146–1157, 2011.

[22] G. Liu, X. Ma, and W. Li, "Publishing node strength distribution with node differential privacy," *IEEE Access*, vol. 8, Article ID 217650, 2020.

[23] Q. Qian, Z. Li, P. Zhao, W. Chen, H. Yin, and L. Zhao, "Publishing graph node strength histogram with edge differential privacy," in *Proceedings of the Database Systems for Advanced Applications*, pp. 75–91, Gold Coast, QLD, Australia, May 2018.

[24] Z. Yuxuan, W. Jianghong, and L. Ji, "Graph degree histogram publication method with node-differential privacy," *Journal of Computer Research and Development*, vol. 56, no. 3, pp. 508–520, 2019.

[25] W. ZhenQiang, H. Jing, and T. Y. Pan, "Privacy preserving algorithms of uncertain graphs in social networks," *Journal of Software*, vol. 30, no. 4, pp. 1106–1120, 2019.

[26] Y. Sun, H. Zhao, Q. Han, and L. Li, "Composite graph publication considering important data," in *Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators*, Changsha, China, September 2017.

[27] A. Haeberlen, B. Pierce, and A. Narayan, "Differential privacy under fire," in *Proceedings of the in 20th USENIX Security Symposium*, San Francisco, CA, USA, August 2011.

[28] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Theory of Cryptography*, vol. 3876, pp. 265–284, 2006.

[29] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 94–103, Providence, RI, USA, November 2007.

[30] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," in *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, pp. 169–178, Miami Beach, FL, USA, December 2009.

[31] J. Leskovec and A. Krevl, *SNAP datasets: stanford large network dataset collection*, 2014, http://snap.stanford.edu/data.

[32] S. Chen and S. Zhou, "Recursive mechanism: towards node differential privacy and unrestricted joins [full version, draft 0.1]," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pp. 653–664, New York, NY, USA, June 2013.

[33] W. Y. Day, N. Li, and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *Proceedings of the 2016 International Conference on Management of Data*, pp. 123–138, San Francisco, CA, USA, June 2016.